

Математическая модель и спецификация интеллектуальных агентных систем

Д. Ю. Бугайченко
arhangel@tepkom.ru

В данной работе развивается метод логической спецификации различных свойств мультиагентных систем, позволяющий в явном виде формализовать понятие «действие», механизмы недетерминированного поведения внешней среды и кооперации агентов. Основу предложенного метода составляет соединение возможностей динамической логики *PDL* и логики альтернированного времени *ATL*, а также адаптация их для спецификации свойств одной из математических моделей интеллектуального агента. Чтобы проиллюстрировать возможности метода, мы приводим пример его применения для спецификации простой мультиагентной системы.

Введение

В последнее время для разработки сложных композитных, в том числе и распределенных, интеллектуальных систем все чаще применяется декомпозиция системы в набор высокоуровневых, автономных, взаимодействующих сущностей — *агентов*. Получаемые в итоге системы составляют широкий класс *мультиагентных систем*.

Своей популярностью мультиагентные системы обязаны основным свойствам, заложенным в концепцию *агента*.

- Агент помещен во *внешнюю среду*, с которой он способен взаимодействовать, но не способен ее контролировать. Следовательно, агент всегда должен быть готов к тому, что его действия не приведут к желаемому результату.

© Д. Ю. Бугайченко, 2006.

- При взаимодействии с внешней средой агент совершает автономные¹ действия, направленные на достижение определенных целей, демонстрируя *управляемое целями поведение*.
- Агент способен адекватно и оперативно реагировать на изменения во внешней среде, другими словами, он обладает свойством *реактивности*.
- Агент способен взаимодействовать с другими сущностями внешней среды, в том числе и с другими агентами, совершая совместные действия для достижения как индивидуальных, так и общих целей.

Таким образом, изолированные интеллектуальные агенты и мультиагентные системы могут успешно применяться при разработке различных открытых систем, интеллектуальных игр, а также систем управления устройствами и комплексами. Для более близкого знакомства с концепцией интеллектуального агента мы рекомендуем работу [17], а информацию об области применения мультиагентных систем можно найти в работах [7] и [16].

При разработке системы управления устройствами и комплексами остро встает вопрос надежности полученной системы, а также ее соответствия исходной задаче. Ошибки в подобных системах могут привести к тяжелым последствиям, вплоть до повреждения дорогостоящего оборудования и даже гибели людей. Для повышения надежности таких систем давно и успешно применяются методы формальной проверки соответствия системы исходной спецификации.

Естественно, для реализации подобных методов необходимо иметь достаточно мощный формальный аппарат, основу которого, как правило, составляет некоторый вариант темпоральной логики. Хороший обзор различных вариантов темпоральной логики можно найти в работе [5].

Различные вариации темпоральной логики неоднократно применялись для формальной спецификации мультиагентных систем, например, в системе DESIRE [4] или языке Dribble [14]. Однако основанные на классической темпоральной логике методы обладают существенным недостатком, затрудняющим их применение для

¹Под автономностью понимается действие без принуждения со стороны внешней управляющей сущности.

спецификации мультиагентных систем — они не предоставляют инструментов для описания кооперации и соперничества агентов, а также для описания недетерминированного поведения внешней среды.

Серьезно ситуация в этой области изменилась всего несколько лет назад с появлением сразу двух работ: коалиционной логики Паули (*Coalition Logic, CL* [8]) и логики альтернированного времени (*Alternating-time Temporal Logic, ATL* [10]). Предложенные логики позволили явно описывать возможности агентов и групп агентов (*коалиций*) по достижению определенных целей вне зависимости от действий агентов вне коалиций и возможной² реакции внешней среды.

Эти работы сразу же привлекли внимание научного сообщества разработчиков мультиагентных систем и получили развитие. Многими авторами [15, 9] было предложено использовать *ATL* в качестве эпистемологической логики (логики для представления знаний агента), а также для формализации концепции *роли* [12]. Интересное развитие получила и коалиционная логика *CL* в работах [2] и [3].

Новые подходы позволили адекватно описывать кооперацию и конкуренцию агентов, однако большинство предложенных подходов по-прежнему не позволяли явно описывать взаимодействие с внешней средой. В этом отношении особо выделяется работа [13]. Предложенная авторами логика позволяет в явном виде описывать возможную реакцию внешней среды на совместные действие коалиций агентов, а также описывать возможности агентов и коалиций по влиянию на состояние внешней среды. Эта логика представляет собой комбинацию пропозициональной динамической логики *PDL* [6] для описания возможной реакции внешней среды и коалиционной логики *CL* для описания возможностей агентов и коалиций. Именно эта работа и вдохновила нас к разработке предлагаемого метода логической спецификации мультиагентных систем.

Постановка задачи

В рамках данной работы мы предложим метод, устраняющий следующие существенные недостатки исходной логики.

²При условии, что реакция внешней среды недетерминирована, но подчиняется некоторым законам, позволяющим определить возможную реакцию на те или иные действия мультиагентной системы.

- При проверке, выполняется ли заданная формула логики (спецификация) на заданной модели, можно получить ответ на вопрос: «*Может ли* система вести себя в соответствии со спецификацией?» В то же время проверить, что система *действительно* ведет себя в соответствии со спецификацией, невозможно.
- Комбинация *PDL* с *CL* ограничивает возможности описания спецификации только одним шагом взаимодействия и не позволяет формулировать утверждения о более сложных сценариях. Источником этого недостатка является ограничение выразительной мощности *CL* одной темпоральной модальностью «в следующий момент».
- Спецификация возможного поведения внешней среды смешивается со спецификацией свойств системы, в результате чего происходит смешение синтаксиса двух логик, что затрудняет использование существующих для этих логик инструментов и алгоритмов.

Для устранения первого недостатка мы предлагаем использовать математическую модель агента, описанную в [17] и расширенную в [1]. Именно для спецификации свойств реализаций этой математической модели и будет применен наш метод.

Расширить выразительную мощность мы предлагаем комбинируя *PDL* не с *CL*, а с *ATL*, которая включает такие темпоральные модальности, как «всегда», «иногда», «до тех пор».

Чтобы избежать смешения синтаксиса двух логик, мы предлагаем иной способ их комбинирования, основная идея которого заключается в использовании спецификации поведения внешней среды в качестве части модели для интерпретации формул свойств системы, наравне с математической моделью самой мультиагентной системы.

Структура работы

В разделе 1 мы приведем краткое определение математической модели интеллектуального агента, которая является предметом спецификации предлагаемой логики.

В основе предлагаемого метода будет лежать классическая логика высказываний, описывающая *состояния* внешней среды и

включающая набор утверждений, интерпретируемых на состоянии внешней среды. Эту логику мы кратко опишем в разделе 2.

Для решения поставленной задачи мы расширим эту логику в двух направлениях:

- для описания *поведения* внешней среды в разделе 3 мы введем ограниченный вариант *PDL* с исходной логикой высказываний в качестве основы;
- для описания свойств системы в разделе 4 мы введем вариант *ATL* с исходной логикой высказываний в качестве основы и формулой логики описания *поведения* внешней среды в качестве части модели.

В разделе 5 мы приведем пример применения метода для спецификации простой мультиагентной системы, а затем подведем итог проделанной работы.

1. Математическая модель интеллектуального агента

В качестве предмета анализа для предлагаемой нами логики будут выступать объекты, описываемые математической моделью агента с состоянием, предложенной в работе [1], поэтому описание логики мы начнем с краткого определения этой модели.

Мультиагентная система включает конечный непустой набор агентов $AG = \{ag_1, \dots, ag_n\}$, которые функционируют в единой внешней среде с конечным непустым множеством состояний S . Каждый из агентов системы представляет собой набор $ag = \{ACA_{ag}, I_{ag}, action_{ag}, refine_{ag}\}$, где:

- $ACA_{ag} = \{aca_{ag,1}, \dots, aca_{ag,m_{ag}}\}$ есть конечное непустое множество действий для данного агента;
- I_{ag} есть конечное непустое множество внутренних состояний данного агента;
- $action_{ag} : I_{ag} \rightarrow ACA_{ag}$ есть функция выбора действия, сопоставляющая каждому внутреннему состоянию определенное действие для данного агента;

- $refine_{ag} : I_{ag} \times S \times ACA_{ag} \rightarrow I_{ag}$ есть функция обновления внутреннего состояния, сопоставляющая предыдущему внутреннему состоянию, новому состоянию внешней среды и действию, совершенному агентом, новое внутреннее состояние.

Общее действие, совершаемое мультиагентной системой, будет представлять собой вектор действий каждого отдельного агента, т. е. $ACS = \{aca_{ag_1}, \dots, aca_{ag_n} \mid aca_{ag_i} \in ACA_{ag_i}\}$. Участие конкретного агента $ag \in AG$ в общем действии системы $acs \in ACS$ мы будем обозначать с помощью записи $acs[ag] \in ACA_{ag}$.

Система взаимодействует с внешней средой, реакция которой на действия системы является недетерминированной и описывается функцией $env : S \times ACS \rightarrow 2^S \setminus \emptyset$. Таким образом, для каждого состояния внешней среды $s \in S$ и для каждого действия системы $acs \in ACS$ определено непустое подмножество состояний, описывающих потенциальный эффект действия.

2. Описание состояния внешней среды

Для описания *состояния* внешней среды (элементов множества S) введем алфавит пропозициональных символов *Prop*. В этом случае множество синтаксически верных утверждений о состоянии внешней среды будет описываться грамматикой

$$\phi ::= p \mid \phi \wedge \phi \mid \neg\phi, \quad (1)$$

где $p \in Prop$.³ Множество всех синтаксически верных утверждений о состоянии внешней среды мы будем обозначать L_{es} .

Для интерпретации пропозициональных символов введем отображение $\pi : S \rightarrow 2^{Prop}$, которое определяет, какие пропозициональные символы истинны для данного состояния внешней среды. Это отображение и будет являться моделью для интерпретации формул описания состояния внешней среды: $M_{es} = \{\pi\}$. Семантика формул описания состояния внешней среды будет описываться с помощью отношения $M_{es}, s \models_{es} \phi$, удовлетворяющего следующим семантическим правилам:

$$1) \ M_{es}, s \models_{es} p \text{ тогда и только тогда, когда } p \in \pi(s);$$

³Здесь и далее мы будем описывать синтаксис и семантику только для логических связок \neg и \wedge , подразумевая, что остальные связки можно выразить классическим способом: $\phi \vee \psi = \neg(\neg\phi \wedge \neg\psi)$, $\phi \Rightarrow \psi = \neg\phi \vee \psi$ и $\phi \Leftrightarrow \psi = (\phi \Rightarrow \psi) \wedge (\psi \Rightarrow \phi)$.

- 2) $M_{es}, s \models_{es} \neg\phi$ тогда и только тогда, когда $M_{es}, s \not\models_{es} \phi$;
- 3) $M_{es}, s \models_{es} \phi \wedge \psi$ тогда и только тогда, когда $M_{es}, s \models_{es} \phi$ и $M_{es}, s \models_{es} \psi$.

Первое семантическое правило описывает интерпретацию пропозициональных символов для заданного состояния внешней среды, а второе и третье являются стандартными для логических связей.

3. Описание поведения внешней среды

Для спецификации поведения внешней среды (свойств функции env) мы введем ограниченный вариант пропозициональной динамической логики PDL с логикой описания состояния внешней среды в качестве основы.

3.1. Синтаксис

Под поведением внешней среды мы понимаем возможную ее реакцию на те или иные действия системы. Для описания действий системы введем алфавит *символов действий*, сопоставив каждому действию каждого агента символ $aca_{ag,i} \sim aca_{ag,i} \in ACS_{ag}$. Множество синтаксически верных формул описания действия будут даваться следующей грамматикой:

$$\alpha ::= aca_{ag,i} \mid \alpha \wedge \alpha \mid \neg\alpha \quad (2)$$

Для описания поведения внешней среды мы используем описанный ранее алфавит пропозициональных символов $Prop$, в этом случае множество синтаксически верных утверждений о поведении внешней среды будет описываться грамматикой

$$\phi ::= p \mid \phi \wedge \phi \mid \neg\phi \mid [\alpha]\phi \quad (3)$$

где $p \in Prop$, а α есть формула языка описания действий. Множество всех синтаксически верных утверждений о поведении внешней среды мы будем обозначать как L_{eb} .

Таким образом, мы расширили логику высказываний (1) дополнительной синтаксической конструкцией $[\alpha]\phi$, интуитивная интерпретация которой звучит так: «При выполнении системой действия,

удовлетворяющего условию α , внешняя среда перейдет в одно из состояний, удовлетворяющих условию ϕ ».

3.2. Семантика

Семантику для формул описания действий мы опишем через отношение $acs \models_{ac} \alpha$ между действием системы $acs \in ACS$ и выражением языка описания действий. Интуитивная интерпретация этого отношения — «действие системы acs удовлетворяет условию α », а формальная интерпретация описывается следующими семантическими правилами:

- 1) $acs \models_{ac} aca_{ag,i}$ тогда и только тогда, когда $acs[ag] = aca_{ag,i}$;
- 2) $acs \models_{ac} \neg\alpha$ тогда и только тогда, когда $acs \not\models_{ac} \alpha$;
- 3) $acs \models_{ac} \alpha \wedge \beta$ тогда и только тогда, когда $acs \models_{ac} \alpha$ и $acs \models_{ac} \beta$.

Первое семантическое правило описывает интерпретацию отдельного символа действия и имеет следующую интуитивную интерпретацию: «Действие системы acs удовлетворяет условию $aca_{ag,i}$, если агент ag выполнил действие $aca_{ag,i}$ ». Второе и третье правила являются стандартными для логических связей.

Для формальной интерпретации формул описания поведения внешней среды мы расширим модель классической логики высказываний M_{es} до пары $M_{eb} = \{env, \pi\}$, где env есть функция, описывающая поведение внешней среды, а π есть интерпретация пропозициональных символов на конкретном состоянии внешней среды.

Семантика формул описания внешней среды будет описываться с помощью отношения $M_{eb}, s \models_{eb} \phi$ между моделью M_{eb} , состоянием внешней среды $s \in S$ и формулой языка спецификации поведения внешней среды $\phi \in L_{eb}$, удовлетворяющего следующим семантическим правилам:

- 1) $M_{eb}, s \models_{eb} p$ тогда и только тогда, когда $p \in \pi(s)$;
- 2) $M_{eb}, s \models_{eb} \neg\phi$ тогда и только тогда, когда $M_{eb}, s \not\models_{eb} \phi$;
- 3) $M_{eb}, s \models_{eb} \phi \wedge \psi$ тогда и только тогда, когда $M_{eb}, s \models_{eb} \phi$ и $M_{eb}, s \models_{eb} \psi$;
- 4) $M_{eb}, s \models_{eb} [\alpha]\phi$ тогда и только тогда, когда для любых $acs \in ACS$ и $s' \in S$ если $acs \models_{ac} \alpha$ и $s' \in env(s, acs)$ то $M_{eb}, s' \models_{eb} \phi$.

Первое семантическое правило описывает интерпретацию позиционных символов, второе и третье являются стандартными для логических связок. Наибольший интерес представляет четвертое семантическое правило, имеющее следующую интуитивную интерпретацию: «Если система в состоянии внешней среды s выполнит действие, удовлетворяющее условию α , то внешняя среда перейдет в одно из состояний, удовлетворяющих условию ϕ ».

4. Описание свойств системы

Для спецификации свойств системы мы используем вариант логики альтернированного времени ATL с логикой описания состояния внешней среды в качестве основы и формулой логики описания поведения внешней среды в качестве части модели. Эта логика позволяет специфицировать свойства математической модели мультиагентной системы, введенной в разделе 1, включая свойства отдельных агентов и групп агентов системы.

4.1. Синтаксис

Введем алфавит *символов коалиций агентов*, сопоставив каждой возможной коалиции (включая пустую коалицию) определенный символ $A \sim A \in 2^{AG}$. В этом случае множество синтаксически верных формул языка описания свойств системы будет задаваться грамматикой

$$\phi ::= p \mid \phi \wedge \phi \mid \neg\phi \mid \langle\langle A \rangle\rangle\bigcirc\phi \mid \langle\langle A \rangle\rangle\Box\phi \mid \langle\langle A \rangle\rangle\phi \mathcal{U}\phi, \quad (4)$$

где $p \in Prop$. Таким образом, язык описания свойств системы расширяет язык описания состояния внешней среды рядом дополнительных конструкций. Рассмотрим их интуитивную интерпретацию:

- конструкция $\langle\langle A \rangle\rangle\bigcirc\phi$ интуитивно интерпретируется как «коалиция A , действуя по программе, переводит внешнюю среду в состояние, удовлетворяющее условию ϕ »;
- конструкция $\langle\langle A \rangle\rangle\Box\phi$ интуитивно интерпретируется как «коалиция A , действуя по программе, добьется того, что внешняя среда всегда будет в состояниях, удовлетворяющих условию ϕ »;

- конструкция $\langle\langle A \rangle\rangle\phi \mathcal{U}\psi$ интуитивно интерпретируется как «коалиция A , действуя по программе, добьется того, что когда-нибудь внешняя среда перейдет в состояние, удовлетворяющее условию ψ , при этом до этого момента внешняя среда будет находиться в состояниях, удовлетворяющих условию ϕ ».

Кроме того, можно описать дополнительный оператор $\langle\langle A \rangle\rangle\Diamond\phi \triangleq \langle\langle A \rangle\rangle true \mathcal{U}\phi$, имеющий интуитивную интерпретацию «коалиция A , действуя по программе, добьется того, что когда-нибудь внешняя среда перейдет в состояние, удовлетворяющее условию ϕ ».

Будем называть формулу языка описания свойств системы формулой с *корректной вложенностью*, если любое вхождение квантора $\langle\langle B \rangle\rangle$ либо свободно, либо находится в области действия квантора $\langle\langle A \rangle\rangle$ и $B \subseteq A$. Значимость этого ограничения мы рассмотрим при описании семантики логики. Далее множество всех синтаксически верных формул языка описания свойств системы с корректной вложенностью мы будем обозначать как L_{MAS} .

4.2. Семантика

Для упрощения дальнейших выкладок введем ряд дополнительных обозначений. Выражение, уникально идентифицирующее действие системы, обозначим как $[acs] \triangleq \bigwedge_{ag \in AG} acs[ag]$, а выражение,

уникально (с точки зрения логики) идентифицирующее состояние внешней среды, обозначим как $[s] \triangleq \bigwedge_{p \in \pi(s)} p \wedge \bigwedge_{q \notin \pi(s)} \neg q$.

На практике, как правило, невозможно описать поведение внешней среды точно и полно в виде функции переходов $env : ACS \times S \rightarrow S$. Тем не менее возможно сформулировать ряд ограничений, которым внешняя среда должна обязательно удовлетворять, составив *спецификацию внешней среды* \widetilde{env} .

Спецификация внешней среды представляет собой набор формул языка описания поведения внешней среды $\widetilde{env} = \{\phi \mid \phi \in L_{eb}\}$, описывающий возможные переходы внешней среды. Обозначим конъюнкцию всех утверждений спецификации внешней среды через $[\widetilde{env}] \triangleq \bigwedge_{\phi \in \widetilde{env}} \phi$.

Заметим, что спецификация внешней среды порождает функцию переходов на множестве состояний внешней среды:

$$\widetilde{env}(acs, s) = S \setminus \{s' \mid \vdash_{eb} \neg([\widetilde{env}] \wedge ([s] \Rightarrow [[acs]][s'])))\}. \quad (5)$$

Таким образом, при совершении системой действия acs состояние внешней среды s может смениться любым состоянием \tilde{s} , если это изменение не противоречит спецификации \widetilde{env} (формула $[\widetilde{env}] \wedge ([s] \Rightarrow [[acs]][s'])$ не является противоречивой).⁴

Моделью для интерпретации формул описания свойств системы будет набор $M = \{AG, \widetilde{env}, \pi\}$, где AG есть множество агентов, \widetilde{env} есть спецификация внешней среды, а $\pi : S \rightarrow 2^{Prop}$ есть интерпретация пропозициональных символов на состояниях внешней среды.

Для формулирования семантических правил нам необходимо описать несколько вспомогательных объектов, порождаемых моделью.

Множество *полных относительно коалиции* $A = \{ag_{j_1}, \dots, ag_{j_k}\} \subseteq AG$ состояний системы определим как декартово произведение множества состояний внешней среды и множеств внутренних состояний для каждого входящего в коалицию агента:

$$FS_A = S \times I_{ag_{j_1}} \times \dots \times I_{ag_{j_k}}. \quad (6)$$

Заметим, что в случае, если коалиция B является подмножеством коалиции A ($B \subseteq A$), возможно естественным образом построить проекцию полного относительно коалиции A состояния в полное относительно коалиции B состояние, исключив внутренние состояния агентов, не входящих в B . Такую проекцию мы будем обозначать как $fs_A|_B$.

Введем вспомогательные обозначения

$$fs_A[ag_{j_i}] = (s, i_{ag_{j_1}}, \dots, i_{ag_{j_k}})[ag_{j_i}] \triangleq i_{ag_{j_i}}$$

для выделения внутреннего состояния одного из агентов коалиции и

$$fs_A[env] = (s, i_{ag_{j_1}}, \dots, i_{ag_{j_k}})[env] \triangleq s$$

для выделения состояния внешней среды.

⁴Логика спецификации поведения внешней среды является ограниченным вариантом логики PDL и, следовательно, задача проверки общезначимости разрешима за экспоненциальное время. Пример полной системы аксиом для данной логики можно найти в [13].

Функцию выбора действия коалиции A в полном относительно этой коалиции состоянии $action_A : FS_A \rightarrow 2^{ACS}$ определим как

$$action_A(fs_A) = \{acs \in A \mid acs[ag] = action(fs_A[ag]), ag \in A\}. \quad (7)$$

Таким образом, выбор действия для агентов, входящих в коалицию, определяется их собственными функциями выбора действия, а поведение агентов вне коалиции недетерминировано.

Функцию переходов полного относительно коалиции состояния системы $trans_A : ACS \times FS_A \rightarrow 2^{FS_A}$ определим следующим образом:

$$trans_A(acs, (s, i_{ag_{j_1}}, \dots, i_{ag_{j_k}})) = \left\{ \left(\begin{array}{l} s', \\ i'_{ag_{j_1}}, \\ \dots, \\ i'_{ag_{j_k}} \end{array} \right) \mid \begin{array}{l} s' \in \widetilde{env}(acs, s), \\ i'_{ag_{j_1}} = refine_{ag_{j_1}}(i_{ag_{j_1}}, s', acs[ag_{j_1}]), \\ \dots, \\ i'_{ag_{j_k}} = refine_{ag_{j_k}}(i_{ag_{j_k}}, s', acs[ag_{j_k}]). \end{array} \right\} \quad (8)$$

Заметим, что недетерминированным является только изменение состояния внешней среды, тогда как изменение внутренних состояний агентов коалиции определяется уже детерминированным образом на основе *измененного* состояния внешней среды.

Определим множество *возможных результатов* действий коалиции агентов A в начальном состоянии fs_A как множество непустых конечных цепочек, полных относительно коалиции состояний системы:

$$out_A(fs_A) = \{\lambda \mid \lambda \in FS_A^+\}, \quad (9)$$

где для каждой цепочки $\lambda[0] = fs_A$, а также для любого $0 \leq i < |\lambda| - 1$ и любого действия системы, удовлетворяющему условию $acs \in action_A(\lambda[i])$, выполнено $\lambda[i+1] \in trans_A(acs, \lambda[i])$.⁵

Теперь мы можем сформулировать семантические правила для интерпретации отношения $M, A, fs_A \models \phi$ между моделью $M = \{AG, \widetilde{env}, \pi\}$, коалицией агентов $A \subseteq AG$, полным относительно коалиции A состоянием системы $fs_A \in FS_A$, и формулой языка описания свойств системы с корректной вложенностью $\phi \in L_{MAS}$. Интуитивная интерпретация этого отношения: «В модели M коалиция A при начальном полном относительно коалиции состоянии

⁵Под обозначением $\lambda[i]$ мы в данном случае понимаем i -й элемент цепочки, а под $|\lambda|$ — длину цепочки.

fs_A ведет себя таким образом, что выполнено свойство ϕ . Формально семантика описывается следующими правилами:

- 1) $M, A, fs_A \models p$ тогда и только тогда, когда $p \in \pi(fs_A[env])$;
- 2) $M, A, fs_A \models \neg\phi$ тогда и только тогда, когда $M, A, fs_A \not\models \phi$;
- 3) $M, A, fs_A \models \phi \wedge \psi$ тогда и только тогда, когда $M, A, fs_A \models \phi$ и $M, A, fs_A \models \psi$;
- 4) $M, A, fs_A \models \langle\langle B \rangle\rangle \bigcirc \phi$ тогда и только тогда, когда $B \subseteq A$ и для любого $acs \in action_B(fs_A|_B)$ и любого $fs'_B \in trans_B(acs, fs_A|_B)$ выполнено $M, B, fs'_B \models \phi$;
- 5) $M, A, fs_A \models \langle\langle B \rangle\rangle \square \phi$ тогда и только тогда, когда $B \subseteq A$ и для любой цепочки $\lambda \in out_B(fs_A|_B)$ и для любого $0 \leq i < |\lambda|$ выполнено $M, B, \lambda[i] \models \phi$;
- 6) $M, A, fs \models \phi \mathcal{U} \psi$ тогда и только тогда, когда $B \subseteq A$ и существует такое $i \geq 0$, что для любой цепочки $\lambda \in out_B(fs_A|_B)$ если $|\lambda| > i$, то $M, B, \lambda[i] \models \psi$ и для любых $0 \leq j < i$ выполнено $M, B, \lambda[j] \models \phi$, а если $|\lambda| \leq i$, то для любых $0 \leq j < |\lambda|$ выполнено $M, B, \lambda[j] \models \phi$.

Заметим, что при описании поведения некоторой коалиции только агенты из этой коалиции ведут себя в соответствии со своей программой, другие же агенты могут выполнять любые действия. Именно из этого семантического нюанса и вытекает ограничение на корректность вложенности формулы ϕ . Если в какой-то момент мы исключаем агента agi из коалиции, то предполагаем, что его действия могут не определяться программой, следовательно, мы не можем знать, в каком внутреннем состоянии находится данный агент и не можем больше определить, как он должен действовать.

Исключение некоторых агентов из коалиции позволяет специфицировать свойства, связанные с отказоустойчивостью системы и конкуренцией агентов. Коалиция должна добиваться своих целей, даже если один из агентов вне её «выйдет из строя» и перестанет вести себя в соответствии с программой, а также в том случае, если агенты вне коалиции будут преследовать собственные цели.

При этом в таком варианте логики нет возможности описать свойства, связанные с присоединением к коалиции нового агента, так как нет возможности определить, в каком внутреннем состоянии он присоединился.

Задача проверки выполнимости формулы на модели для используемой нами логики является алгоритмически разрешимой для конечных моделей. Алгоритмическая сложность этой задачи соответствует сложности аналогичной задачи для логики ATL , т. е. определяется полиномом от числа возможных переходов системы. Более подробную информацию об алгоритмической сложности ATL можно получить в [10].

5. Пример

Для иллюстрации предложенных концепций рассмотрим пример мультиагентной системы управления автономным исследовательским зондом, оснащённым солнечными батареями и аккумуляторами, шасси, позволяющим перемещение и повороты на месте, а также манипулятором, позволяющим поднимать предметы из окружающего мира.

Любая выполняемая зондом операция ведет к расходу энергии и может истощить аккумуляторы, чего управляющая система не должна допустить. Для предотвращения разрядки аккумуляторов зонд может использовать солнечные батареи — если батареи раскрыты, то аккумулятор автоматически заряжается, но только в светлое время суток.

Некоторые явления внешней среды могут оказать вредное воздействие на солнечные батареи, для предотвращения этого зонд должен сворачивать солнечные батареи, после чего они перестанут заряжать аккумулятор.

При перемещениях по внешнему миру зонд может встретить препятствия, столкновений с которыми он должен избегать. Чтобы избежать повреждения манипулятора, зонд должен поднимать предметы только определенных размеров и не пользоваться манипулятором во время движения и операций с солнечными батареями.

5.1. Спецификация возможных действий

Систему управления таким зондом логично компоновать из трех агентов:

- управления солнечными батареями SB с действиями: открыть батареи $open$, закрыть батареи $close$ и ничего не делать $noop_{SB}$;

- управления движением MV с действиями: переместиться вперед $move$, повернуться $turn$ и ничего не делать $noop_{MV}$;
- управления манипулятором MN с действиями: поднять предмет $pickup$ и ничего не делать $noop_{MN}$.

5.2. Спецификация внешней среды

Далее мы перейдем к описанию состояния внешней среды мультиагентной системы. Заметим, что к внешней среде будет относиться как окружающий зонд мир, так и собственная аппаратура зонда. Введем следующие пропозиции:

- состояние батарей описывает пропозиция $batteryOpen$, интерпретируемая как истинная в том случае, если батареи открыты;
- для описания состояния аккумулятора используем три взаимоисключающих пропозиции $chargeGood$, $chargeMedium$ и $chargeExhausted$;
- пропозицией $isSunny$ будем обозначать тот факт, что в данный момент солнца достаточно для зарядки батарей;
- наличие опасных для солнечных батарей явлений будем обозначать пропозицией $danger$;
- для описания окружающей зонд обстановки будем использовать две взаимоисключающих пропозиции $obstacle$ и $sample$. Первая пропозиция указывает на то, что непосредственно перед зондом находится препятствие, а вторая на то, что непосредственно перед зондом находится предмет, который можно поднять;
- факт того, что зонд подобрал некоторый полезный предмет из внешнего мира, мы будем обозначать с помощью пропозиции $collected$, а того, что зонд переместился, пропозицией $moved$;
- факт того, что зонд получил повреждения, мы будем обозначать пропозициями $hullDamaged$, $armDamaged$ и $batteryDamaged$. Первая пропозиция будет обозначать повреждение корпуса, вторая будет обозначать повреждение манипулятора, а третья — повреждение солнечных батарей.

Формализуем ранее описанные закономерности, используя логику спецификации поведения внешней среды. Для начала опишем закономерности, связанные с зарядом аккумулятора и солнечными батареями.

- $(chargeGood \wedge \neg chargeMedium \wedge \neg chargeExhausted) \vee (\neg chargeGood \wedge chargeMedium \wedge \neg chargeExhausted) \vee (\neg chargeGood \wedge \neg chargeMedium \wedge chargeExhausted)$ — это утверждение описывает определенность состояния аккумулятора в каждый конкретный момент времени.
- $(batteryOpen \wedge isSunny) \Rightarrow (chargeGood \Rightarrow [true]chargeGood) \wedge (chargeMedium \Rightarrow [true]chargeMedium \vee chargeGood)$ — это утверждение описывает закономерности изменения заряда аккумулятора. В условиях наличия солнца и открытых батарей действия агентов не ведут к ухудшению заряда аккумулятора.
- $(chargeGood \Rightarrow [(noop_{SB} \vee open) \wedge noop_{MV} \wedge noop_{MN}]chargeGood) \wedge (chargeMedium \Rightarrow [(noop_{SB} \vee open) \wedge noop_{MV} \wedge noop_{MN}]chargeMedium)$ — операция открытия батарей или полное бездействие не ведет к ухудшению заряда аккумулятора.
- $(batteryOpen \Rightarrow [close \wedge noop_{MV} \wedge noop_{MN}]\neg batteryOpen \wedge \neg batteryDamaged) \wedge (\neg batteryOpen \wedge \neg danger \Rightarrow [open \wedge noop_{MV} \wedge noop_{MN}]batteryOpen \wedge \neg batteryDamaged)$ — это утверждение описывает последствия операций открытия или закрытия батарей, которые выполнялись в условиях отсутствия других параллельных действий. Заметим, что безопасность операции открытия батарей обеспечена только при отсутствии опасных явлений во внешнем мире.
- $\neg danger \vee \neg batteryOpen \Rightarrow [\neg((close \vee open) \wedge (move \vee turn \vee pickup))]\neg batteryDamaged$ — при отсутствии опасных явлений во внешнем мире или в условиях закрытых батарей действия, не являющиеся комбинацией операций над батареями с другими действиями, не ведут к повреждению батарей.

Рассмотрим закономерности, связанные с перемещениями агента и работой с манипулятором:

- $\neg(\text{obstacle} \wedge \text{sample})$ — это простое утверждение о том, что перед зондом не может находиться одновременно и препятствие, и предмет, который можно поднять.
- $(\neg\text{obstacle} \Rightarrow [\text{move}](-\text{hullDamaged} \wedge \text{moved})) \wedge [\neg\text{move}]-\text{hullDamaged}$ — это утверждение описывает безопасные условия для перемещения зонда. Движение при отсутствии препятствий и любая комбинация действий без движения не ведет к повреждению корпуса.
- $(\text{sample} \Rightarrow [\text{pickup} \wedge \text{noop}_{MV}]-\text{armDamaged} \wedge \text{collected}) \wedge [\neg\text{pickup}]-\text{armDamaged}$ — если перед зондом находится предмет, который можно поднять, то выполнение действия *pickup* приведет к тому, что будет подобран полезный предмет, если зонд не попытается переместиться или развернуться. Кроме того, только действие *pickup* может повредить манипулятор.
- $[\neg\text{pickup}]-\text{collected}$ — только действие *pickup* может привести к тому, что будет подобран полезный предмет из внешнего мира.

Заметим, что при спецификации внешней среды мы описывали ситуации, ведущие к положительным эффектам или хотя бы к отсутствию негативного эффекта. Далее, при генерации функции переходов негативный эффект от перехода будет возможен всегда, кроме тех случаев, когда спецификация явно говорит, что такого перехода быть не может.

5.3. Спецификация свойств системы

В этом разделе мы приведем примеры формализации некоторых свойств системы, состав, возможности и среду для которой мы описали выше. Как правило, при спецификации мультиагентной системы наибольший интерес представляют *инвариантные* свойства, то есть свойства, выполненные на протяжении всей жизни системы.

Для начала рассмотрим формулировку некоторых наиболее простых свойств, заключающихся в отсутствии негативных событий.

- $\langle\langle\{SB, MV, MN\}\rangle\rangle \Box \neg(\text{chargeExhausted} \vee \text{batteryDamaged})$ — это свойство требует того, чтобы солнечные батареи зонда

всегда были в порядке, а аккумулятор не был разряжен. Добиться выполнения этого свойства может только вся система в целом.

- $\langle\langle\{MV, MN\}\rangle\rangle \Box \neg\text{armDamaged}$ — это свойство требует того, чтобы манипулятор системы всегда не был поврежден. Добиться этого система должна даже в том случае, если агент *SB* выйдет из строя.
- $\langle\langle\{MV\}\rangle\rangle \Box \neg\text{hullDamaged}$ — это свойство требует того, чтобы корпус зонда всегда оставался в порядке. Добиться этого система должна даже в том случае, если нормально функционировать будет только агент *MV*.

Все приведенные свойства для описываемой системы могут быть выполнены при правильном и согласованном поведении агентов. При этом наиболее сложно достижение первого свойства, так как для этого необходимы согласованные действия именно всех агентов. Свойство вида

$$\langle\langle\{SB, MV\}\rangle\rangle \Box \neg(\text{chargeExhausted} \vee \text{batteryDamaged})$$

уже не может быть выполнено в описываемых условиях, так как действие агента *MN*, не согласованное с действиями других агентов, может привести к повреждению солнечных батарей или разряду аккумулятора.

Еще одним часто встречающимся классом свойств мультиагентной системы, являются свойства достижимости позитивных результатов, например.

$$\langle\langle\{SB, MV, MN\}\rangle\rangle \Diamond \text{collected}.$$

Это свойство требует того, чтобы зонд когда-нибудь подобрал хотя бы один предмет.

В описанных нами условиях построение системы, гарантированно удовлетворяющей этому свойству, невозможно, так как наличие предмета перед зондом является необходимым условием для подбора предмета, но это наличие не контролируется системой.

5.4. Пример системы

В заключение мы приведем пример простой мультиагентной системы и рассмотрим задачу проверки соответствия её спецификации. Для реализации агентов системы мы используем простую производящую систему, что не требует поддержки внутреннего состояния.

Агент SB активируется только при изменении условий освещенности или наличия опасных явлений. Его задача заключается в том, чтобы закрывать солнечные батареи в случае опасности или отсутствия солнца и открывать солнечные батареи в противном случае. Правила для агента SB будут иметь вид:

- 1) $(danger \vee \neg isSunny) \wedge batteryOpen \rightarrow close;$
- 2) $isSunny \wedge \neg danger \wedge \neg batteryOpen \rightarrow open;$
- 3) $true \rightarrow noops_B.$

Агент MN должен воздерживаться от действий, если солнечные батареи закрыты, отсутствует солнце или наблюдаются опасные явления, и поднимать встречающиеся на пути предметы, если батареи открыты, светит солнце и опасность батареям не угрожает. Правила для агента MN будут иметь вид:

- 1) $danger \vee \neg isSunny \vee \neg batteryOpen \rightarrow noop_{MN};$
- 2) $sample \rightarrow pickup;$
- 3) $true \rightarrow noop_{MN}.$

Агент MV должен воздерживаться от действий в случаях, если батареи закрыты или должны быть закрыты, а также, если перед зондом находится предмет, и перемещаться по внешнему миру, если батареи открыты. Правила для агента MV будут иметь вид:

- 1) $danger \vee \neg isSunny \vee \neg batteryOpen \vee sample \rightarrow noop_{MV};$
- 2) $\neg obstacle \rightarrow move;$
- 3) $true \rightarrow turn.$

Решение задачи проверки соответствия спецификации $\langle\langle\{MV, MN\}\rangle\rangle \Box \neg armDamaged;$ $\langle\langle\{MV\}\rangle\rangle \Box \neg hullDamaged;$ $\langle\langle\{SB, MV, MN\}\rangle\rangle \Box \neg (chargeExhausted \vee batteryDamaged)$ для приведенной системы выдаст множество состояний внешней среды, удовлетворяющих условию $\neg (chargeExhausted \vee batteryDamaged \vee hullDamaged \vee armDamaged) \wedge (\neg batteryOpen \vee \neg danger \vee isSunny \vee chargerGood)$. Начав взаимодействовать с внешней средой в начальном состоянии, удовлетворяющем этому условию, и ведя себя в соответствии с программой, система будет удовлетворять спецификации. Таким образом, система может избежать негативных последствий для зонда, если зонд изначально был в хорошем состоянии и условиях.

Если же мы к спецификации системы добавим свойство $\langle\langle\{SB, MV, MN\}\rangle\rangle \Diamond collected$, то на множество начальных состояний внешней среды при решении задачи проверки соответствия добавится дополнительное ограничение $sample \wedge isSunny \wedge \neg danger \wedge batteryOpen \wedge chargeGood$. Таким образом, гарантировать подбор хотя бы одного образца можно только в том случае, если система сразу попадает в условия, когда этот подбор возможен.

Заключение

В последнее время проблеме построения формального метода для спецификации различных аспектов мультиагентных систем уделяется все больше внимания в современном научном сообществе. Во многом это связано с появлением коалиционной логики [8] и логики альтернированного времени [10], которые позволили решить многие проблемы, связанные с многокомпонентностью мультиагентных систем.

Коалиционная логика получила интересное развитие в работе [13], где авторы, используя комбинацию коалиционной логики с пропозициональной динамической логикой [6], смогли формализовать взаимодействие мультиагентной системы с внешней средой и возможности мультиагентной системы.

В нашей работе мы предложили метод решения таких открытых вопросов, как спецификация более сложных темпоральных свойств, задача проверки соответствия системы спецификации и разделения синтаксиса двух логик.

Для спецификации более сложных темпоральных свойств мы предложили комбинировать пропозициональную динамическую ло-

гику с логикой альтернированного времени, чьи возможности богаче, чем у коалиционной логики.

Разделения синтаксиса двух логик (пропозициональной динамической логики и логики альтернированного времени) мы добились с помощью включения спецификации поведения внешней среды в *модель* для логики спецификации свойств системы. Разделение синтаксиса двух логик позволяет использовать существующие инструменты и алгоритмы для решения основных задач в предложенной логике спецификаций мультиагентных систем.

В качестве наиболее интересных направлений развития предложенной темы мы видим практические примеры адаптации инструментов для *ATL*, например *МОСНА* [11], к решению задачи проверки соответствия системы спецификации и автоматического синтеза системы по спецификации, а также расширение логики возможностями описания свойств не только состояния среды, но и внутренних состояний агентов. Последнее может привести к расширению модальных операторов логики для рассуждения о представлениях, желаниях и целях агентов.

Список литературы

- [1] Бугайченко Д. Ю., Соловьев Д. Ю. Абстрактная архитектура интеллектуального агента и методы её реализации // Системное программирование. Вып 1. СПб.: Изд-во СПбГУ, 2005. С. 36–67.
- [2] Agotnes T., Van der Hoek W., Wooldridge M. On the Logic of Coalitional Games // AAMAS '06: Proc. Fifth International Joint Conference on Autonomous Agents and Multiagent Systems. 2006. P. 153–160.
- [3] Agotnes T., Van der Hoek W., Wooldridge M. Temporal Qualitative Coalitional Games // Ibid. P. 177–184.
- [4] Brazier F. M. T., Dunin-Keplicz B. M., Jennings B. M., Treur J. DE-SIRE: Modelling Multi-Agent Systems in a Compositional Formal Framework // Int. Journal of Cooperative Information System. 1997. P. 67–94.
- [5] Emerson E. A. Temporal and Modal Logic. Amsterdam: North-Holland Pub. Co., 1990. 77 p.
- [6] Harel D., Kozen D., Tiuryn J. Dynamic logic // SIGACT News. 2001. P. 66–69.
- [7] Jennings N. R., Wooldridge M. J. Applications of Intelligent Agents. London: Queen Mary & Westfield College, University of London, 2000. 27 p.
- [8] Pauly M. Logical for Social Software. PhD thesis, Amsterdam. 2001. 193 p.
- [9] Pauly M., Wooldridge M. Logic for mechanism design — a manifesto // GTDT-2003: Proc. 2003 Workshop on Game Theory and Decision Theory in Agent Systems. 2003.
- [10] Rajeev A., Henzinger T. A., Kupferman O. Alternating-time Temporal Logic // J. ACM. 2002. P. 672–713.
- [11] Rajeev A, Henzinger T. A., Mang F. Y, C. et al. Mocha: Modularity in model checking // LNCS 1427. 1998. P. 521–525.
- [12] Ryan M., Schobbens P.-Y. Agents and roles: refinement in alternating-time temporal logic // ATAL-2001: Intelligent AgentsVIII: Proc. Eighth International Workshop on Agent Theories, Architectures, and Languages. 2001. P. 100–114.
- [13] Sauro L., Gerbrandy J., Van der Hoek W., Wooldridge M. Reasoning about Action and Cooperation // AAMAS '06: Proc. Fifth International Joint Conference on Autonomous Agents and Multiagent Systems. 2006. P. 185–192.
- [14] Van Riemsdijk B., Van der Hoek W., Meyer J.-J. Ch. Agent programming in dribble: from beliefs to goals using plans // AAMAS'03: Proc. Second International Joint Conference on Autonomous Agents and Multiagent Systems. 2003. P. 393–400.
- [15] Van der Hoek W., Wooldridge M. Cooperation, Knowledge, and Time: Alternating-Time Temporal Epistemic Logic and Its Applications // Studia Logica. 2003. P. 125–157.
- [16] Van Dyke Parunak H. Industrial and Practical Application of DAI // Multiagent Systems. 2001. P. 27–79.
- [17] Wooldridge M. J. Intelligent Agents // Multiagent Systems. 2001. P. 27–79.